



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/620,615	07/17/2003	Hiroshi Oyama	116597	9663
25944	7590	08/13/2008		
OLIFF & BERRIDGE, PLC P.O. BOX 320850 ALEXANDRIA, VA 22320-4850			EXAMINER PRICE, NATHAN E	
			ART UNIT 2194	PAPER NUMBER
			MAIL DATE 08/13/2008	DELIVERY MODE PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/620,615	<b>Applicant(s)</b> OYAMA ET AL.	
	<b>Examiner</b> NATHAN PRICE	<b>Art Unit</b> 2194	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 12 June 2008.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-6 and 8-14 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-6 and 8-14 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)          | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____                                      |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____  | 6) <input type="checkbox"/> Other: _____                          |

### **DETAILED ACTION**

1. Claims 1 – 6 and 8 – 14 are pending.
2. This Office Action is in response to communications received 12 June 2008.  
Previous objections and rejections not included in this Office Action have been withdrawn.

### ***Response to Arguments***

3. Applicant's arguments with respect to claims 1 – 6 and 8 – 14 have been considered but are moot in view of the new ground(s) of rejection.

### ***Claim Rejections - 35 USC § 112***

The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 1 – 6 and 8 – 14 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. It appears that the original

disclosure does not support the IDLs using the same language to specify the claimed details of each function as now recited in claims 1 – 6 and 8. It also appears that the original disclosure does not support the device register being identified by a function as now recited in the independent claims. The disclosure supports a register that identifies a function (specification p. 6 lines 14 – 18).

### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1 – 6 and 8 – 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schofield (US 6,308,225 B1) in view of Umar (see PTO-892 mailed with this Office Action) and Silberschatz (see PTO-892 mailed 20 March 2007 and 20 June 2007).

6. As to claim 1, Schofield teaches an interface method for a logical circuit comprising a logical operation element, comprising:

defining an interface, using a first interface definition language which is partly common to a second interface definition language directed to a software object such that a part of the first interface definition language and a part of the

second interface definition language use the same language to specify an interface name, a function name, and an argument and a return value, wherein the first interface definition language has means for defining a plurality of functions, each function having a function name and a function return value, and at least one function having at least one function argument [col. 2 lines 45 - 65; col. 3 lines 12 - 32, 54 - 65; col. 5 lines 6 - 13; col. 8 lines 12 - 22; col. 12 lines 9 - 18];

providing at least means for inputting for identifying the function name defined by the first interface definition language for a server interface circuit in order to realize the interface among the means for inputting for identifying the function name defined by the first interface definition language, means for inputting and outputting the argument, and means for outputting the return value [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22];

determining whether the function is the at least one function having the at least one function argument [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22; col. 12 lines 6 - 43]; and

performing at least one of inputting the function argument, outputting the function argument and outputting the function return value [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22; col. 12 lines 6 - 43],

wherein the software object is capable of realizing the interface independently of the server interface circuit by using only the second interface definition language [col. 3 lines 20 - 30; col. 4 lines 8 - 18].

7. Although Schofield teaches return values for operations, it does not clearly state that return values are defined for each individual function. However, Umar teaches signatures being defined for each individual function and provides an example of an IDL that defines return values for each function (p. 309 ¶1 – 2; Table 7.1). It would have been obvious to one of ordinary skill in the art at the time Applicant's invention was made to combine these teachings because Schofield teaches IDL for CORBA (col. 3 lines 12 – 17) and Umar teaches additional details about IDL for systems including CORBA (p. 309 ¶2).

8. Schofield fails to specifically teach defining a hardware interface or identifying a device register as claimed. However, Silberschatz teaches a server process interacting with a device for a client process [page 470 ¶ 1 – 3] and that objects include hardware objects [page 569 ¶ 1]. This results in the server process acting as an interface to the hardware. Silberschatz also teaches the hardware interface includes a function identifying a device register [page 404 ¶2]. It would have been obvious to one of ordinary skill in the art at the time Applicant's invention was made to combine these teachings because both teach use of client-server systems and Silberschatz provides additional details about what can be represented as an object and how clients and other software can interface with various objects in a computer system.

9. As to claim 2, Schofield teaches an interface method for a logical circuit comprising a logical operation element, comprising:

defining an interface, using a first interface definition language which is partly common to a second interface definition language directed to a software object such that a part of the first interface definition language and a part of the second interface definition language use the same language to specify an interface name, a function name, and an argument and a return value, wherein the first interface definition language has means for defining a plurality of functions, each function having a function name and a function return value, and at least one function having at least one function argument [col. 2 lines 45 - 65; col. 3 lines 12 - 32, 54 - 65; col. 5 lines 6 - 13; col. 8 lines 12 - 22; col. 12 lines 9 - 18];

providing at least means for inputting for identifying the function name defined by the first interface definition language for a client interface circuit in order to realize the interface among the means for outputting for identifying the function name defined by the first interface definition language, means for inputting and outputting the argument, and means for inputting the return value [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22];

determining whether the function is the at least one function having the at least one function argument [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22; col. 12 lines 6 - 43]; and

performing at least one of inputting the function argument, outputting the function argument and outputting the function return value [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22; col. 12 lines 6 - 43],

wherein the software object is capable of realizing the interface independently of the server interface circuit by using only the second interface definition language [col. 3 lines 20 – 30; col. 4 lines 8 – 18].

10. Although Schofield teaches return values for operations, it does not clearly state that return values are defined for each individual function. However, Umar teaches signatures being defined for each individual function and provides an example of an IDL that defines return values for each function (p. 309 ¶1 – 2; Table 7.1).

11. Schofield fails to specifically teach defining a hardware interface. However, Silberschatz teaches a server process interacting with a device for a client process [page 470 ¶ 1 – 3] and that objects include hardware objects [page 569 ¶ 1]. This results in the server process acting as an interface to the hardware. Silberschatz also teaches the hardware interface includes a function identifying a device register [page 404 ¶2].

12. As to claim 3, Schofield teaches an interface method for a logical circuit comprising a logical operation element, comprising:



defining an interface, using a first interface definition language which is partly common to a second interface definition language directed to a software object such that a part of the first interface definition language and a part of the second interface definition language use the same language to specify an interface name, a function name, and an argument and a return value, wherein the first interface definition language has means for defining a plurality of functions, each function having a function name and a function return value, and at least one function having at least one function argument [col. 2 lines 45 - 65; col. 3 lines 12 - 32, 54 - 65; col. 5 lines 6 - 13; col. 8 lines 12 - 22; col. 12 lines 9 - 18] wherein the logical circuit comprises:

a server logical circuit, as a server interface circuit for realizing the interface, having at least means for inputting for identifying the function name defined by the first interface definition language among the means for inputting for identifying the function name defined by the first interface definition language, means for inputting and outputting the argument, and means for outputting the return value [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22], and

a client logical circuit, as a client interface circuit for realizing the interface, having at least means for outputting for identifying the function name defined by the first interface definition language among the means for outputting for identifying the function name defined by the first interface definition language, means for inputting and outputting the argument, and means for inputting the return value [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22], and

data being transferred from the means for outputting for identifying the function name of the client logical circuit to the means for inputting for identifying the function name of the server logical circuit [col. 3 line 54 - col. 4 line 7],

the server logical circuit and the client logical circuit each having at least one of the means for outputting the return value and the means for inputting the return value, and data can be transferred from the means for outputting the return value to the means for inputting a return value [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7];

determining whether the function is the at least one function having the at least one function argument [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22; col. 12 lines 6 - 43]; and

performing at least one of inputting the function argument, outputting the function argument and outputting the function return value [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22; col. 12 lines 6 - 43],

wherein the software object is capable of realizing the interface independently of the server interface circuit by using only the second interface definition language [col. 3 lines 20 - 30; col. 4 lines 8 - 18].

13. Although Schofield teaches return values for operations, it does not clearly state that return values are defined for each individual function. However, Umar teaches signatures being defined for each individual function and provides an example of an IDL that defines return values for each function (p. 309 ¶1 - 2; Table 7.1).

14. Schofield fails to specifically teach defining a hardware interface. However, Silberschatz teaches a server process interacting with a device for a client process [page 470 ¶ 1 – 3] and that objects include hardware objects [page 569 ¶ 1]. This results in the server process acting as an interface to the hardware. Silberschatz also teaches the hardware interface includes a function identifying a device register [page 404 ¶2].

15. As to claim 8, Schofield teaches the server logical circuit and the client logical circuit each having the means for inputting and outputting the argument and data being transferred between the means for inputting and outputting the argument of the server logical circuit and means for inputting and outputting the argument of the client logical circuit [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7].

16. As to claim 5, Schofield teaches a device having an interface and a logical circuit, which defines an interface, using a first interface definition language which is partly common to a second interface definition language directed to a software object such that a part of the first interface definition language and a part of the second interface definition language use the same language to specify an interface name, a function name, and an argument and a return value and has means for defining a plurality of functions, each function having a function name and a function return value and at least one function having at least one function argument [col. 2 lines 45 - 65; col. 3 lines 12 - 32, 54 - 65; col. 5 lines 6 - 13; col. 8 lines 12 - 22], comprising:

a client interface circuit for realizing the interface comprises means for outputting for identifying the function name defined by the first interface definition language among the means for outputting for identifying the function name defined by the first interface definition language, means for inputting and outputting the argument, and means for inputting the return value [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22];

an argument number detection section for determining whether the function is the at least one function having the at least one function argument [col. 2 lines 32 – 33; col. 9 lines 51 – 53]; and

wherein the software object is capable of realizing the interface independently of a server interface circuit by using only the second interface definition language [col. 3 lines 20 – 30; col. 4 lines 8 – 18].

17. Although Schofield teaches return values for operations, it does not clearly state that return values are defined for each individual function. However, Umar teaches signatures being defined for each individual function and provides an example of an IDL that defines return values for each function (p. 309 ¶1 – 2; Table 7.1).

18. Schofield fails to specifically teach defining a hardware interface or an argument register as claimed. However, Silberschatz teaches a server process interacting with a device for a client process [page 470 ¶ 1 – 3] and that objects include hardware objects [page 569 ¶ 1]. This results in the server process acting as an interface to the

hardware. Silberschatz also teaches an argument register for inputting the at least one function argument and for outputting the at least one function argument [page 404 ¶2; page 405 ¶2]. Silberschatz also teaches the hardware interface includes a function identifying a device register [page 404 ¶2].

19. As to claim 6, Schofield teaches:

the connection terminal of the client interface is connected to the server interface circuit or a system bus [Schofield: Fig. 1; col. 5 lines 14 – 23], and

when the connection terminal of the client interface is connected to the server interface circuit, the device connected with the server interface circuit is drivable via the server interface circuit [Schofield: Fig. 1; col. 3 line 66 – col. 4 line 7; col. 5 lines 14 – 23], and

when the connection terminal of the client interface is connected to the system bus, a value of the register within the client interface can be read via a central processing device such that the central processing device can serve in the place of the device connected with the server interface circuit [Schofield: Fig. 1; col. 3 line 66 – col. 4 line 7; col. 5 lines 14 – 23] [Silberschatz: page 402 ¶ 6; page 405 ¶ 2].

20. Schofield fails to specifically teach the register as claimed. However, Silberschatz teaches the client interface circuit has a connection terminal and a register the value is stored in the register [Silberschatz: page 402 ¶ 6; page 405 ¶ 2].

21. As to claim 11, Schofield teaches a device having a hardware interface and a logical circuit comprising a logical operation element, which defines an interface, using an interface definition language having means for defining a function name, an argument, and a return value for each function defined by the function name [col. 2 lines 45 - 65; col. 3 lines 12 - 32, 54 - 65; col. 5 lines 6 - 13; col. 8 lines 12 - 22], wherein

a server interface circuit for realizing the interface comprises:

means for inputting for identifying the function name defined by the interface definition language among the means for inputting for identifying the function name defined by the interface definition language, means for inputting and outputting the argument, and means for outputting the return value [col. 2 lines 45 - 65; col. 3 line 54 - col. 4 line 7; col. 8 lines 12 - 22];

an argument number detection section for determining whether the function is the at least one function having the at least one function argument [col. 2 lines 32 - 33; col. 9 lines 51 - 53]; and

a client interface circuit is capable of realizing the interface independently of the server interface circuit by using another interface definition language [col. 3 lines 20 - 30; col. 4 lines 8 - 18].

22. Schofield fails to specifically teach defining a hardware interface or an argument register as claimed. However, Silberschatz teaches a server process interacting with a device for a client process [page 470 ¶ 1 - 3] and that objects include hardware objects

[page 569 ¶ 1]. This results in the server process acting as an interface to the hardware. Silberschatz also teaches an argument register for inputting the at least one function argument and outputting the at least one function argument [page 404 ¶2; page 405 ¶2]. Silberschatz also teaches the hardware interface includes a function identifying a device register [page 404 ¶2].

23. As to claim 4, see the rejection of claim 1.

24. As to claims 9 and 12, see the rejection of claim 2.

25. As to claim 10, see the rejection of claim 3.

26. As to claim 13, see the rejection of claim 6.

27. As to claim 14, see the rejection of claim 8.

### ***Conclusion***

28. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to NATHAN PRICE whose telephone number is (571)272-4196. The examiner can normally be reached on 6:00am - 2:30pm, Monday - Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.



Art Unit: 2194

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/  
Supervisory Patent Examiner, Art Unit 2195

NP